

8-2016

# A Dynamic Hair Rigging System for Maya

Wyndham Batton

Clemson University, wbatton.dpa@gmail.com

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_theses](https://tigerprints.clemson.edu/all_theses)

---

## Recommended Citation

Batton, Wyndham, "A Dynamic Hair Rigging System for Maya" (2016). *All Theses*. 2489.  
[https://tigerprints.clemson.edu/all\\_theses/2489](https://tigerprints.clemson.edu/all_theses/2489)

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

# A DYNAMIC HAIR RIGGING SYSTEM FOR MAYA

---

A Research  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment  
of the Requirements for the Degree  
Masters of Fine Arts  
Digital Production Arts

---

by  
Wyndham Batton  
August 2016

---

Accepted by:  
Dr. Tessendorf, Committee Chair  
Dr. Brian Malloy  
Dr. Joshua Levine

# Abstract

This thesis will cover the creation, use, and process of an automated rigging tool in Maya. The dynamic hair-rigging system's purpose is to help control the function, style, and movement of hair, along with other uses. A solution for controlling and manipulating sections of hair is using an automated rigging system that allows for full artistic freedom over styling and animating hair. An interface was designed to aid rigging artists with producing quick and efficient rigs for hair. Along with the use for hair, the dynamic hair-rigging tool could also be used to drive and control chain, rope, and other secondary objects.

# Table of Contents

<b>Title Page</b> . . . . .	<b>i</b>
<b>Abstract</b> . . . . .	<b>ii</b>
<b>List of Figures</b> . . . . .	<b>iv</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
<b>2 Background and Related Work</b> . . . . .	<b>3</b>
2.1 Maya nHair . . . . .	3
2.2 Rigging . . . . .	4
2.3 Related Work . . . . .	5
<b>3 Interface</b> . . . . .	<b>9</b>
<b>4 Creating the Joints and Controls</b> . . . . .	<b>12</b>
4.1 Curve and Joints . . . . .	12
4.2 Renaming and Selecting Joints . . . . .	13
4.3 Controllers . . . . .	14
<b>5 Dynamics and Creating the Rig</b> . . . . .	<b>16</b>
5.1 Naming . . . . .	16
5.2 Dynamics . . . . .	16
5.3 Extras . . . . .	20
5.4 Creating the Rig . . . . .	21
<b>6 Example Uses/Possibilities</b> . . . . .	<b>24</b>
<b>7 Final Results</b> . . . . .	<b>27</b>
<b>8 Future Applications/Add Ons</b> . . . . .	<b>30</b>
<b>9 Conclusion</b> . . . . .	<b>33</b>
<b>Appendices</b> . . . . .	<b>34</b>
<b>A Hair Rigging Code Snippets:</b> . . . . .	<b>35</b>
<b>Bibliography</b> . . . . .	<b>43</b>

# List of Figures

1.1	Character from DPA Short Animation: Misfit Mice . . . . .	2
2.1	Characters from Disney’s animated film Zootopia . . . . .	6
2.2	Rabbit Character: Judy Hopps from Disney’s animated film Zootopia . . . . .	7
2.3	Merida’s hair process created by Claudia Chung and her team at Pixar . . . . .	7
2.4	Ben Sledge’s Automated Build System used on a jellyfish model . . . . .	8
3.1	Interface of Rigging Tool . . . . .	11
4.1	Creating Joints . . . . .	12
4.2	Selecting the Anatomical Joint Type . . . . .	13
4.3	Renaming the Joints . . . . .	14
4.4	Selecting Joints . . . . .	14
4.5	Selecting Control Number . . . . .	15
5.1	Dynamic Properties . . . . .	19
5.2	Dynamic Property Options in Channel Box . . . . .	20
5.3	Hair System Shape . . . . .	22
5.4	Extra Tool Options . . . . .	23
6.1	Elsa from Disney’s animated film Frozen . . . . .	24
6.2	Sherman from DreamWorks Animation’s animated film Mr. Peabody and Sherman . . . . .	25
6.3	Using Hair Rig to Create Dynamic Chains . . . . .	26
7.1	Hair Rig Bare Bones . . . . .	28
7.2	Hair Rig on Mery Project Model . . . . .	28
7.3	Second pose of Mery with hair rig . . . . .	29
8.1	Using Tool to Create New Locks of Controllable Hair . . . . .	31
8.2	Possibility of having controls influence a wide range of hair by controlling more than one curve. . . . .	32

# Chapter 1

## Introduction

This thesis focuses on automated rigging implementation and stylization of hair and other features using Maya software from Autodesk. The concept of the rigging tool was to create a dynamic rigging process to control sections of hair, along with other uses such as chains, ropes, appendages, etc. A common problem that occurs between computer generated hair and rigging is the constant refinement through multiple iterations. A solution to rigging and hair iterability was to automate the rigging process as much as possible, giving the user artistic freedom and control over the style and animation of hair.

The inspiration for this tool came from a character created for a short animation, Misfit Mice, in the Digital Production Arts program at Clemson University. The final result of this character is displayed in figure 1.1. Since this was the first time the Digital Production Arts program tried to implement fur/hair in a short the resources were limited to tutorials and online papers. After many challenges and hurdles it was decided that Houdini would be used to create the fur for the character. Though more information was provided through Maya regarding hair, Houdini was seen as the better choice since Houdini saves fur descriptions internally and there was no need to cache the descriptions. The Digital Production Arts pipeline was not built to handle Maya fur descriptions, so when a turn table or animation was sent to review, the majority of the data was lost and the character came out missing chunks of fur.

Having little knowledge of dynamics in Houdini, the character's fur in the Digital Production Arts short remained static. Since the character in the short film was a mouse, adding dynamics to a rodent with short fur did not seem necessary. Unlike the goal in the short, which was to create



Figure 1.1: Character from DPA Short Animation: Misfit Mice

photo realistic fur, the goal for this dynamic hair-rigging tool was the creation of a rig that could aid in the styling and movement of hair, chains, ropes, etc. Combining Maya's dynamic hair system with an animatable rig allowed for more control and efficiency when animating a character or prop.

This thesis starts with an overview of rigging and the Maya nHair system along with related works and other hair rigging solutions in Chapter 2. An overview of the dynamic hair-rigging system's interface is discussed in Chapter 3 while a detailed examination of the interfaced is discussed in Chapters 4 and 5. Example hair styles for which the rigging system could be used for are discussed in Chapter 6. Chapter 7 depicts example characters which are rigged using the dynamic hair-rigging system. Finally, Chapters 8 and 9 discuss the future applications and conclusion.

## Chapter 2

# Background and Related Work

One of the most important breakthroughs in creating realism in a character has been the appearance of hair. Computer generated hair has been a difficult task due to the complexity and the computational power needed to render millions of hairs. Companies, such as Pixar, have come up with ways to produce more realistic hair dynamics, styles, overall look, and styling. Problems with computer generated hair can be seen when dynamics are added to the hair system. Without creating custom parameters, hair can be styled and formed to a user's desire but when dynamics is added, it becomes a whole other task. This chapter discusses the details of the Maya nHair system, the process of rigging, and different schemes for rigging hair and controlling hair styles.

### 2.1 Maya nHair

Many methods and tools have been created to get hair to a certain level of detail. An example of this is the Maya nHair System. The nHair system simulates natural movement and collisions of hair, hair underwater, hair being blown in the wind, various hairstyles, and other effects such as ropes, chains, cables, wires, and much more. Maya's nHair uses the Maya Nucleus, the dynamic system that creates nCloth and nParticle simulations. The nHair system has the ability to interact with different Nucleus objects such as nCloth, passive collision, and nParticle objects. In order for these simulations to interact, all of the objects that will be interacting with each other must have the same Nucleus solver. When using nHair in Maya, there is a group created called hair follicles. These follicles make up the hair system, and much like how a human follicle hosts a hair,



Maya follicles host a hair curve. The follicle curves control the attributes and curves associated with a hair clump, as well as how the hairs attach to a surface. Another group available after the hair system is created is the hair system output curves. The output curves consist of NURBS curves, Paint Effects strokes, or both. If the hair follicles' output are NURBS, each follicle will hold one NURBS curve that represents the position of the hair; if Paint Effects, then the follicle will be made up of strokes [3].

## 2.2 Rigging

Rigging is the process of designing and implementing the manipulation architecture for an animated three-dimensional character. During this process, a hierarchal system of movement and articulation control for the animator is built into a 3D model [10]. Terms to know when rigging are joints, inverse kinematics, forward kinematics, control curves, constraints, skinning, and weight painting.

In Maya, joints or bones are used in rigging in the same manner as joints in the human body. The joints are points of articulation created to control the model. When rigging a character, joints are placed in areas that bend or rotate, such as the shoulder, elbow, neck, hips, etc.

Inverse Kinematics (IK) is a system put on a joint chain where the child node in the hierarchy influences/controls the movements of its parents. Animations that benefit from Inverse Kinematics are pushing against a wall, swinging on a bar, or supporting the body with the arm. Having the child node control the movements of the parents allows animator to animate independently of the chain's hierarchy. Unlike Inverse Kinematics, Forward Kinematics (FK) follow the hierarchal chain in the rig. This means there is more control over the chain but also means it will take more time to animate. Each joint would need to be positioned independently starting from the parent joint and moving down the chain. Since FK can take longer to animate than IK, a lot of riggers incorporate both in order to have the most control and have the rig meet the animator's needs.

To help control the IK/FK in the joint chains, control curves are created to assist the animator in manipulating these joints within the rig. Control curves benefit the animator since the joint chains are positioned within a character's mesh. Control curves, made with NURBS curves, are placed around or outside a character so the curves are easy to select and used to position the character rather than the actual joints. To help keep parts of a model from breaking during animation,

constraints can be used to limit the range of motions of a character’s movement. Constraints can limit position, rotation, and scale based off attributes on the parent objects. An example of constraints could be parenting an object to another to have the child follow what the parent is doing. Another example could be putting constraints on a control for the hand to limit how far the joints rotate on the fingers to prevent penetration into the palm.

The last two terms are skinning and weight painting. Skinning is what binds the joints to the 3D mesh or character. Binding the joints to the skin will allow the mesh to move along with the animation of the joints. Going along with skinning, an important step in this process is weight painting. Weight painting influences how much control the skeleton has over the mesh. An example is if an arm joint has too much influence on the mesh it might affect the neck or shoulder and produce unrealistic results [8].

## 2.3 Related Work

Though tools in different software packages have become more and more powerful, especially in the past couple of years, studios and FX teams have relied on creating their own tools to get the look and feel of hair desired by the industry. One example is Disney’s tool iGroom, which was used to create the fur for all the animals in ”Zootopia” shown in Figure 2.1. The fur-controlling tool helped shape around 2.5 million hairs on the lead bunny character Judy Hopps, shown in figure 2.2, and around 9 million on a giraffe. The movie included 64 different species which amassed to 800,000 animal characters [5]. Disney’s new tool gave the animators more flexibility in the creation of fur than ever before. A key feature of this tool is the ability to iterate quickly through each character. Before this tool, FX artists could still create the hair they wanted but the animators had to work with hair proxies. Artists had to predict the way their characters would change after the fur was added. Waiting hours for a render to come back could work with one character but rendering 800,000 characters with fur would be too time consuming [7].



Figure 2.1: Characters from Disney’s animated film Zootopia

Similar to how Disney created a new tool for hair and fur, Claudia Chung and her team at Pixar came up with a way to produce curly hair for the animated film "Brave". The team at Pixar hand placed many different types of springs/curves on the model of Merida, ranging from short, long, thin, fat, bouncy, and stiff. The types of curves placed on Merida can be seen in figure 2.3. The problem with using a spring like structure for hair is the need to keep it stiff enough to hold the curl shape, as well as keeping its soft movement. The technique Chung’s team used to achieve this was called "core curve and points". The system worked much like a pearl or beaded necklace where the curves resembled the string of the necklace while the springs acted like beads. The result was the ability to move Merida’s head and the curls would move along the curves and keep their shape, all while maintaining the look of the character [9].



Figure 2.2: Rabbit Character: Judy Hopps from Disney's animated film Zootopia

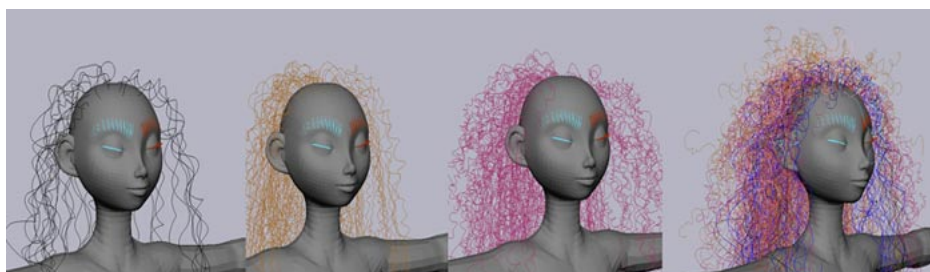


Figure 2.3: Merida's hair process created by Claudia Chung and her team at Pixar

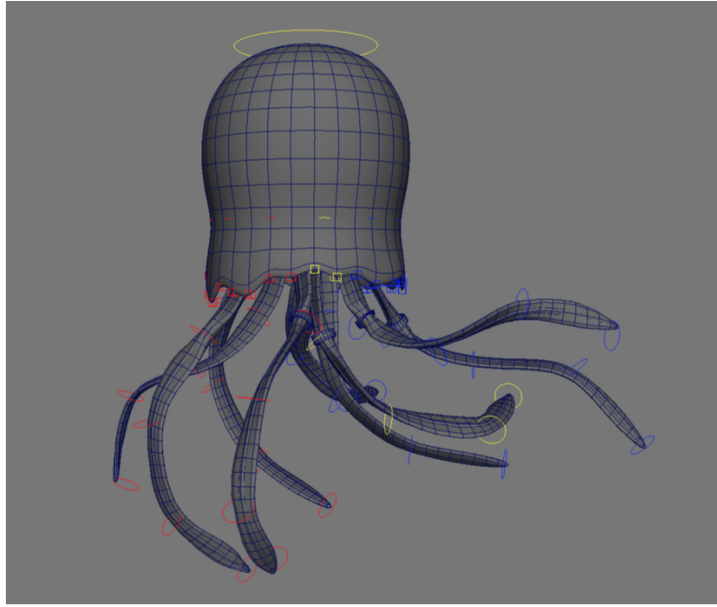


Figure 2.4: Ben Sledge’s Automated Build System used on a jellyfish model

Another related work is Ben Sledge’s automated rigging system documented in his 2014 Thesis, “An Automated Build System for Articulated Characters”. Sledge’s thesis focuses on the time-intensive process of creating complex rigs and proposes a method to produce high quality rig quickly and efficiently. The purpose behind Ben Sledge’s solution was to automate the rigging process as much as possible by giving the rigger predefined module elements (rig blocks) or the ability to script new ones. The script-based nature of the system provides a powerful tool set for rigging artists to quickly apply on a variety of characters [10]. An example of how Ben Sledge’s rigging system in use can be seen in figure 2.4.

Though similar to Sledge’s automated rigging process, the dynamic hair-rigging system only focuses on creating rigs to control hair and objects with similar dynamic qualities. Sledge’s automated build system uses Maya’s nHair for rigging objects such as tentacles, but is the not main focus. Since the automated build system intends for user customization, the script based approach allows users to make changes without affecting the overall rig. Though the dynamic hair-rigging system is GUI-based rather than script based, the system allows artists to quickly apply rigs to a variety of objects. Differing from Sledge’s approach to use Maya nHair solely for animation purposes, the dynamic hair-rigging system allows the ability to manipulate hair animation, as well as the styling and appearance of hair within Maya.

## Chapter 3

# Interface

The dynamic hair-rigging system interface was created using the programming language python. Python was chosen due to the language being powerful, fast, immediate run time, portable, integrated into Maya, and has extensive standard libraries. The purpose behind the interface was to allow a user to create a dynamic rig all in one window rather than search through animation and dynamics shelves in Maya. The interface can be seen in figure 3.1 and will be discussed throughout the paper. Each section of the interface contains processes that are used when rigging a character or other objects in a scene. The steps included in the interface are the creation of joints, renaming joints to fit naming conventions, creating control curves, adding dynamics to the joints, and finally creating the rig.

The first section of the interface allows a user to pick the number of joints used in the rig, the type of joints used, joint names, and controls added to the joints. Joint types were included in the interface to give users the ability to rig models with restricted movement. Restricting movement by joint types rather than restricting movement through locking parameters proved to be convenient and efficient.

The next section includes an area for renaming joints and selecting joints. Adding a section to rename joint chains was important because naming conventions help a user distinguish between areas of a rig. The purpose of selecting joints was to define a joint chain the rig will be attached to by selecting the first and last joints in the chain while including the between joints.

The controller tab of the interface allows the user to choose the number of controls being added to the rig. Adding controls to the rig allows for better maneuverability and control over the

rig rather than controlling the rig directly from the joints. Limitations were set on the amount of controls added to a rig to ensure maneuvering functionality. The user could add as many curves to the rig as long as the number did not exceed the number of joints.

The Dynamic Hair Properties tab includes the main Maya nHair dynamic attributes needed for the rig, as well as adding a prefix to the dynamic system. By default the dynamic attributes are checked, meaning they will be included in the rig. The user has the option to disable dynamics all together or to disable certain dynamic attributes. The options selected to include in the tool are start curve attract, attraction damp, dampening, collision, self collision, gravity, motion drag, drag, mass, stretching, and point lock. Each of the choices listed are discussed further in later chapters.

The last two tabs in the dynamic hair-rigging interface include an extras section and a curve color override section. The features included in the extras are selecting the hair system and baking the simulation. Though certain dynamic attributes are included when the rig is created, a user may need to adjust other attributes in the hair system. When the select hair system is chosen, the full list of the hair system attributes is displayed in the channel box/attribute editor, allowing the user to further customize the dynamics and behavior of the rig. The option to bake the simulation on the curves provided more control on a given simulation. Not every aspect of a simulation/animation will come out perfect so being able to fix minor problems can be adjusted by editing the keyframes produces from baking the simulation. An area this proved helpful was fixing minor penetration issues with collisions between objects. The last section of the rigging tool was added simply to aid in identifying which curves control what part of the rig. Typically sections of controls are grouped together by colors on a character. The ligaments of a character are divided by one side being blue and the other being red while the torso controls are colored yellow or another color.

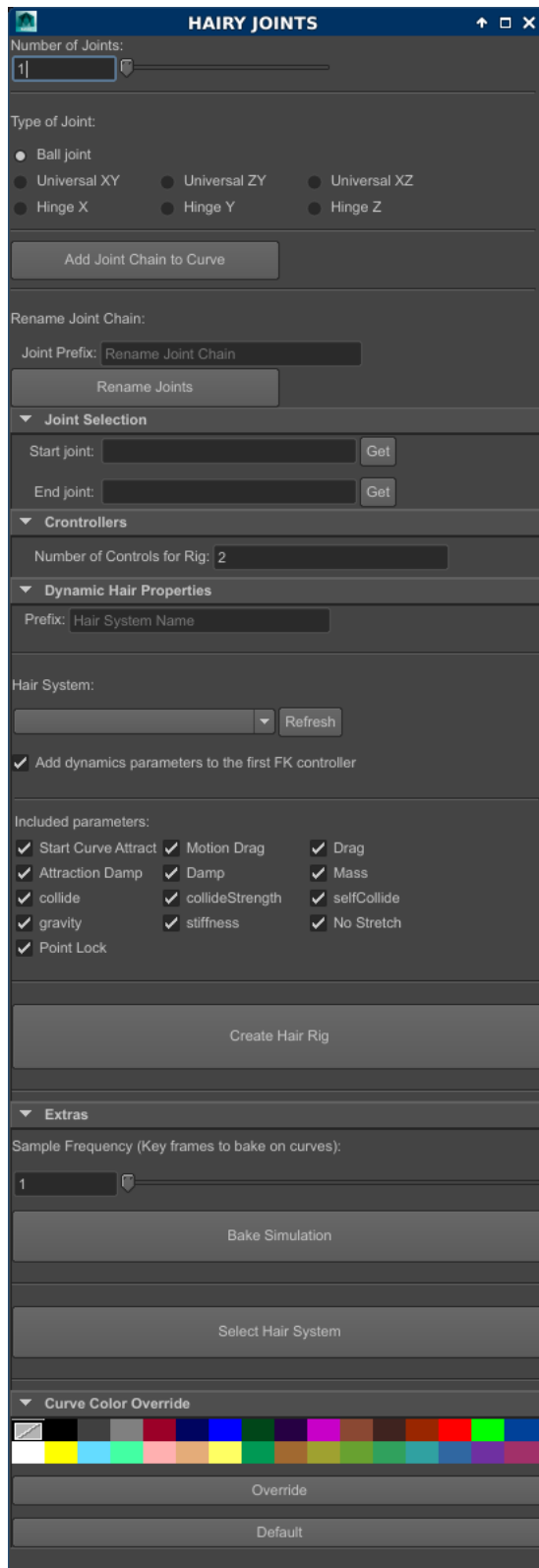


Figure 3.1: Interface of Rigging Tool



## Chapter 4

# Creating the Joints and Controls

### 4.1 Curve and Joints

Joint chains produced by the dynamic hair-rigging tool were intended to be created along a path, specifically along a CV or NURBS curve. The purpose behind this method was to allow a joint chain to be created procedurally for an object being rigged instead of going through the trouble of manually placing joints on a model or curve. Once a curve was created, the next step was to decide how many joints needed to be added to the curve. To decide how many joints to place on a curve a number slider was added to the interface which can be seen in figure 4.1. The number of joints to be created corresponds to the number chosen on the slider.

Along with deciding how many joints were going to be added to a curve another option included was the ability to select between three types of anatomical joints. The choices included ball joint, universal joint, and hinge joint. A ball joint has the ability to rotate along all three local axes with no restrictions making this type of joint ideal for rigging a neck or shoulder of a character. The universal joint is one that only rotates along two of the local axes. An example where the universal joint is ideal is a wrist since a wrist's only movements are up, down, left, and right. A hinge



Figure 4.1: Creating Joints

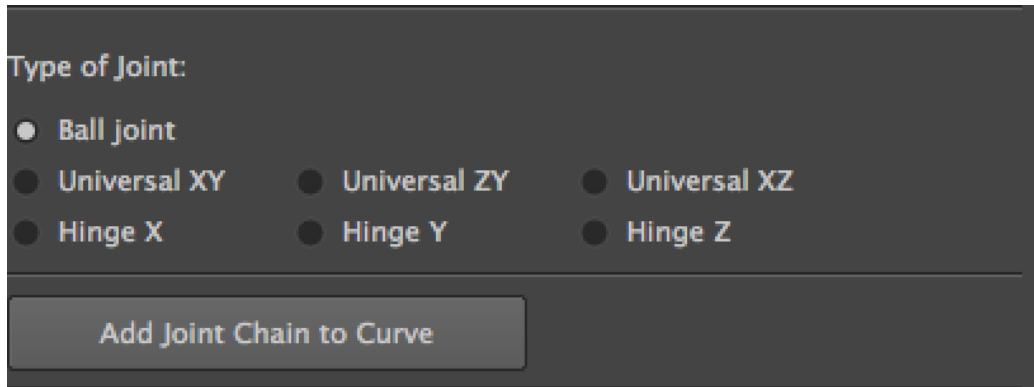


Figure 4.2: Selecting the Anatomical Joint Type

joint’s rotation is restricted to all but one local axes, much like the human knee [2]. If a user were to select a joint type other than a ball joint, the user would need the ability to select which axes to allow the joint to rotate. Displayed in figure 4.2 are the choices in the tool for the types of joints available to a user. Aside from the ball joint, each joint choice was broken down to three different choices. The choices for the universal joint selection include rotations on the XY, XZ, or YZ axes and the choices for the hinge joint include rotations on the X, Y, or Z axis.

## 4.2 Renaming and Selecting Joints

Naming conventions have always been important when labeling entities and keeping a workspace organized. Well chosen naming conventions have helped users in navigating and searching in large structures. This section of dynamic hair rig interface was designed to allow a user to change the default naming of the joints to a description that better reflects what the joints are controlling. Figure 4.3 shows the text box where the user inputs the new description of the joint chain. The purpose of renaming the joints was not to be used to rename joint chains that require different descriptions for each joint, an example being a human arm, but ones that use the same description for each joint. Each joint will be renamed with the same description while keeping the number that corresponded with each joint.

In order for the rig to be created, the dynamic hair-rigging tool needed to know what joints to attach the rig to. To achieve this, a section was added to the interface to retrieve the joints needed to create the rig. As seen in figure 4.4 two different buttons and text boxes were created,

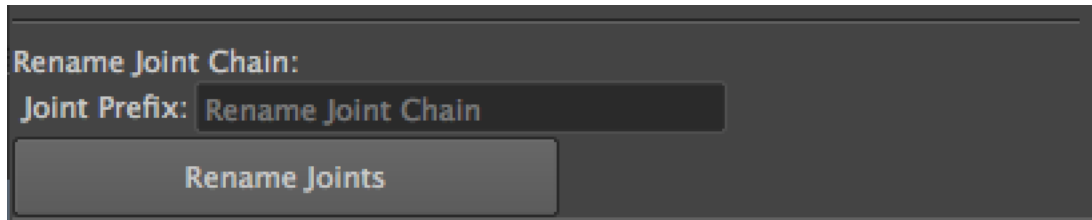


Figure 4.3: Renaming the Joints

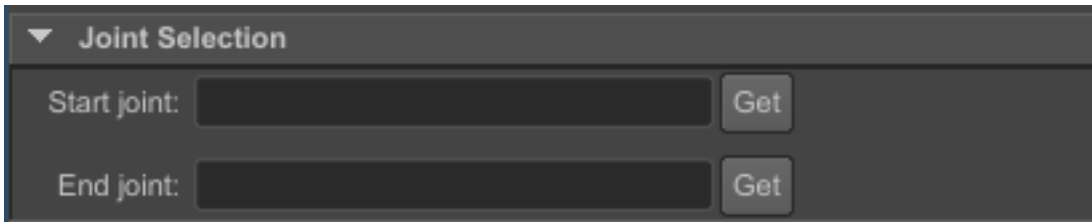


Figure 4.4: Selecting Joints

one to retrieve the first joint and the other to retrieve the last joint of a joint chain. Each button would be pressed to load the joints into the text boxes after the joints were selected.

## 4.3 Controllers

In order to allow the joint chain to be animated without directly controlling the joints, a section to add control curves was included in the interface which can be seen in figure 4.5. The control curves are created from NURBS, which are non-renderable curves or geometry. Adding control curves to the rig allows for posing and animating to be completed with ease by providing visuals for an animator to interact with. Any number of control curves could be added to the rig as long as the amount of controls did not exceed the number of joints. The movement and control over the joints are directly influenced by the number of control curves. Having the same amount of controls as joints resulted in each curve influencing one joint. If the number of control curves are less than the number of joints, the influence of each curve was spread across multiple joints, creating less control over the rig as a whole. The best outcome is having a control curve for each joint.



Figure 4.5: Selecting Control Number

## Chapter 5

# Dynamics and Creating the Rig

### 5.1 Naming

This chapter will cover the dynamics that are added to the rig once it has been created. In figure 5.1 the Dynamic Hair Properties section of the rigging tool is shown. The first part of the dynamic portion added was a space to add a prefix/name to the rig. Since naming conventions are used industry wide and help with organization it was necessary to include the option to name the rigging system. If a scene contained multiple hair systems before the rig has been created, the user could name each rig after the corresponding hair systems or the actual hair systems. As an example for rigged sections of hair, each system could be named bangs, ponytail, strayhair, etc. The next step in this process was to assign a hair system to the rig. If a hair system is already being used in a scene, then hitting the refresh button would load the hair system in the drop down menu and would be able to be selected. If a new hair system was being created with the rig then the menu would be left blank.

### 5.2 Dynamics

Next is adding dynamic options to the first control of the rig, also seen in figure 5.1. Though Maya's nHair system provides many options to customize and make hair more artistically pleasing or accurate, the options on the tool are the main ones needed to get started. The options provided from the dynamic hair-rigging system include:

- **Start Curve Attract:** Determines the amount of attraction of the current hair position to the start position
- **Damp:** Dampens the relative shape change of individual hairs affecting how the hair bends and stretches
- **Attraction Damp:** Damps the effect of Start Curve Attract, decreasing the velocity of hair as it moves towards its start curves positions
- **Motion Drag:** Damps the movement of your hair curves relative to the movement of their follicles
- **Drag:** Simulates friction with the air, in addition to helping to stabilize the simulation
- **Mass:** Sets the simulated mass of the hair vertices
- **Collide:** nHair object collides with Nucleus objects
- **Collide Strength:** Specifies strength of collisions between the nHair object and other Nucleus object
- **Self Collide:** Dynamic hairs generated by the hair object collide with each other
- **Gravity:** Values affect the influence of gravity on the hair system
- **Stiffness:** Affects the overall stiffness from root to tip of the hair
- **Stretching:** Helps create realistic, flexible, non-stretchy hair without having to increase the number of Nucleus Substeps
- **Point Lock:** Specifies where the hair is attached relative to the start position

With the start curve attract, deciding this value can depend on the length of hair or object that it is being applied to. Start curve attract determines the amount of attraction of the hair position to that of the starting position. For long flowing hair the value should be set to or around the value zero, and for short hair the value can range anywhere between zero and one. Dampening affects how the hairs bend and stretch while the attraction damp decreases the velocity of the hair as it moves closer to the starting curve position. The attraction damp can be a useful attribute since it will lessen the spring of the hair if the attraction value for the hair is set to a high value. Motion

drag is self-explanatory; it dampens the movement of the curves/hair relative to the movement of the hair follicles. For example, if the motion drag is set to one, the hair will appear as if it is moving through molasses. One way to make the hair seem realistic however is adjusting the stiffness scale to where there is less motion drag the further away from the follicle the hair is. Collision is another option available for the rig. In order to have realistic hair, the hair should be able to collide with the body and with each hair strand. The strength of the collisions is also a key factor since it specifies the strength of the collisions between the hair object and the other nucleus objects. Having the value of zero is the same as having no collisions while one is having full collision. A problem that can occur if the value is set to one is the objects getting stuck or trapped during collision. A solution to this is to lower the value of the collision strength. Instead of having to dig through the outliner to find the hair system, these dynamic options, if chosen to be applied, are available in the channel box editor when the first control of the rig is highlighted. In this tool, the user can decide to dismiss the dynamic parameters completely or pick which ones will be included. One thing to note with adding hair systems is the fact that a passive collider can only be assigned one solver so if multiple hair systems are within in a scene only one of them can interact with a specific object [1].

The last item added to the rig is the point lock, which is originally part of the follicle shape. The point lock specifies where the hair is attached relative to the start position. There are four different positions to attach the hair. The first option is not attaching the hair at all. If this choice is made the hair will fall when the simulation is ran. This could be useful if a character is loosing hair or if another object needs to fall with gravity. The next option is attaching the hair at the base. Constraining the base is what is used for actual hair. The next option is constraining the tip. This option can be used to a user's preference depending on what the rig is being used for. When the simulation is ran with the option the chain acts the same way as if it were constrained at the base just the opposite side. The last option is constraining both ends. This could be useful when adding the dynamic rig to a rope or chain that is attached to something at either end. When the simulation is ran with this option both ends will stay stationary while the rest of the chain runs through the simulation. Though the follicle shape node options are No Attach, Base, Tip, and BothEnds, the point lock attribute in the FK curve values are 0,1,2, and 3. Each of these values corresponds to the choices from the follicle point lock. If the user chooses 0 then the hair will not be constrained on either end and so forth.

▼ **Dynamic Hair Properties**

Prefix:

Hair System:

▼

☒ Add dynamics parameters to the first FK controller

Included parameters:

<input checked="" type="checkbox"/> Start Curve Attraction	<input checked="" type="checkbox"/> Motion Drag	<input checked="" type="checkbox"/> Drag
<input checked="" type="checkbox"/> Attraction Damp	<input checked="" type="checkbox"/> Damp	<input checked="" type="checkbox"/> Mass
<input checked="" type="checkbox"/> collide	<input checked="" type="checkbox"/> collideStrength	<input checked="" type="checkbox"/> selfCollide
<input checked="" type="checkbox"/> gravity	<input checked="" type="checkbox"/> stiffness	<input checked="" type="checkbox"/> No Stretch
<input checked="" type="checkbox"/> Point Lock		

Figure 5.1: Dynamic Properties



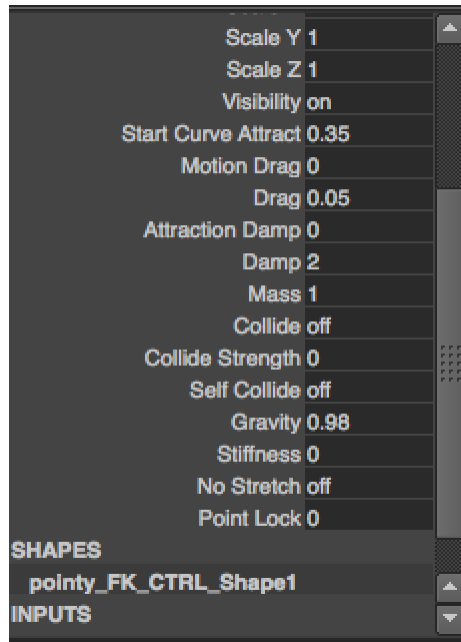


Figure 5.2: Dynamic Property Options in Channel Box

### 5.3 Extras

Figure 5.2 shows the dynamics added from the tool and figure 5.3 shows other options that are available with this tool, one being Select Hair System. As stated before the last two sections included in the dynamic rig interface were an extras section and a curve color override section which can be seen in figure 5.4. The features included in Extras are selecting the hair system and baking the simulation. Though certain dynamic attributes were included when the rig was created, a user may need to adjust other attributes in the hair system. When the select hair system button is pressed, the full list of the hair system attributes are displayed in the channel box/attribute editor, allowing the user to further customize the dynamics and behavior of the rig.

The option to bake the simulation on the curves provides more control on a given simulation. Not every aspect of a simulation/animation will be perfect, so having the ability to edit keyframes allows for problem areas to be fixed. This is helpful for fixing minor penetration issues with collisions between objects. The tool iterates through a selected group and bakes the simulation on each curve within that group. The slider above the Bake Simulation button allows a user to adjust the sampling frequency of the bake. If the value is set at "1", the simulation will run with a sampling frequency

of one time unit, meaning for every frame in the sequence a key frame will be placed. If the value is set to "2" the simulation will run with a sampling frequency of two time units, meaning for every two frames a key frame will be placed. The benefit from this is to be able to scrub through the timeline and fix any problems that occur by changing the values of the keyed frame. If there are any collision or movement problems that cannot be fixed by changing the simulation, editing the key frames is a way to fix the issues.

The last tab of the rigging tool was added to aid in identifying which curves control each section of the rig. Typically sections of controls are grouped together by colors on a character. The ligaments of a character are divided by one side being blue and the other being red while the torso controls are typically yellow. This option was added to assign a color to the hair controls, specifically the controller with the dynamic properties assigned to it, allowing the user to know which control curve to choose if changes need to be made. There are 32 color options available and once a control and color are selected, clicking the override button will change that control's color. Hitting the default button will set the color back to its original [11].

## 5.4 Creating the Rig

Once the rig has been created, a couple of things will be added to the outliner. These include a Forward Kinematic joint chain group which drives the joints, a control group which contains the control curves used to control rig positions, and if a new hair system is created a dynamic group will be added, which contains the hair system, nucleus, follicles, and output curves. Similar to grouping joints and curves when rigging a character, to have the rig work properly the FK joint group will need to be parented to the same group the joints and the control curves group parented under the controls they are suppose to follow.

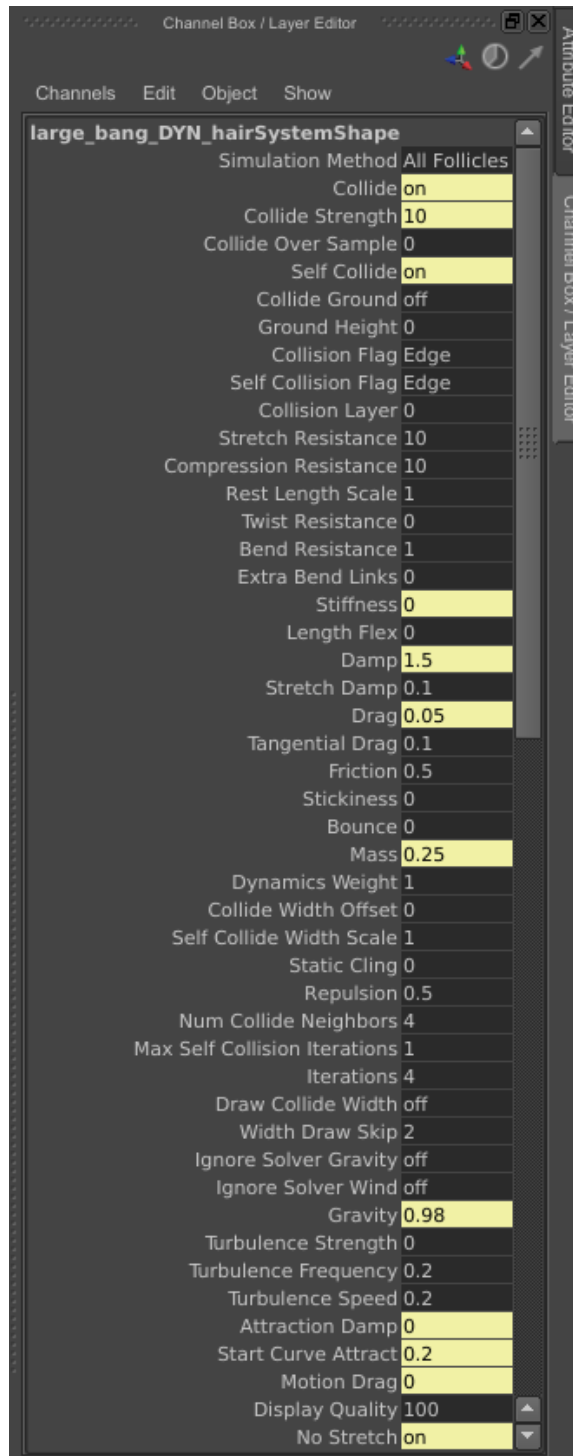


Figure 5.3: Hair System Shape

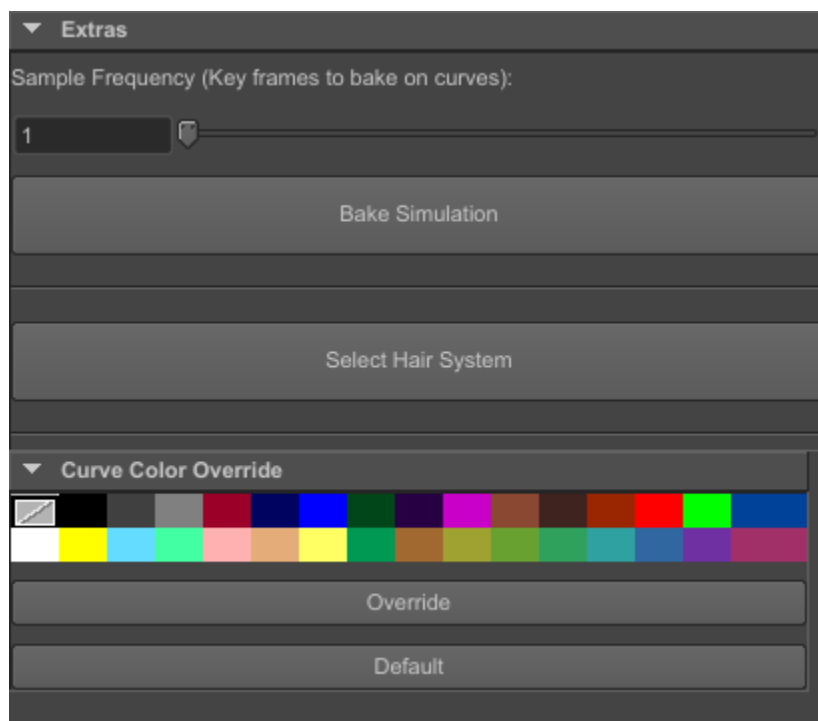


Figure 5.4: Extra Tool Options

## Chapter 6

# Example Uses/Possibilities

An example of a hair style that could be created with the rigging tool would be Elsa's hairstyle from the film, Frozen. As seen in figure 6.1 Elsa's hair style, though seemingly realistic, was broken into large sections or clumps. The type of hair style could be ideal for the dynamic hair rigging tool since the amount of curves created from the hair system would be limited. Instead of having added constraints to hold the hair on top of Elsa's hair, the rigs created from the dynamic hair rigging tool could be point locked on each side of the curve allowing for the hair to move realistically while staying locked on her forehead and along the base of the ponytail. The movement of the pony tail could be accomplished with this tool as well.



Figure 6.1: Elsa from Disney's animated film Frozen



Figure 6.2: Sherman from DreamWorks Animation’s animated film Mr. Peabody and Sherman

Another example of hair the dynamic hair-rigging tool could control is that of Sherman from DreamWorks Animation, Mr. Peabody and Sherman. In the style of the film, Sherman’s hair was not photo realistic but still came across as hair as seen in figure 6.2. With the way his hair was designed, the rigging tool would not require many guide hairs in order to move the hair as needed. When looking that this approach there could be a need for part of the rig such as a main controller. The user could have the controls of the rig on each guide curve but could also have a main controller that would move most or all the hair at once. Another possibility is to have rig controls as a part of that main controller. With this, a user could move the entire head of hair around but also manipulate and reshape the entire mass of hair as well [4].

In figure 6.3, the chain, is an example of another use of this tool. Though a more rigorous tool and options for creating a chain may be needed if the chain was a "hero" object. Since this tool focuses on the dynamic movement of hair, the movements of a chain will be limited. Unlike a hair strand, a chain is made up of links that are connected and have a limited amount of freedom to move within each link so this tool would not be able to simulate a balled up or jumbled chain. This tool would be useful for chains on a draw bridge, prison restraints, background objects, anchors, and more.

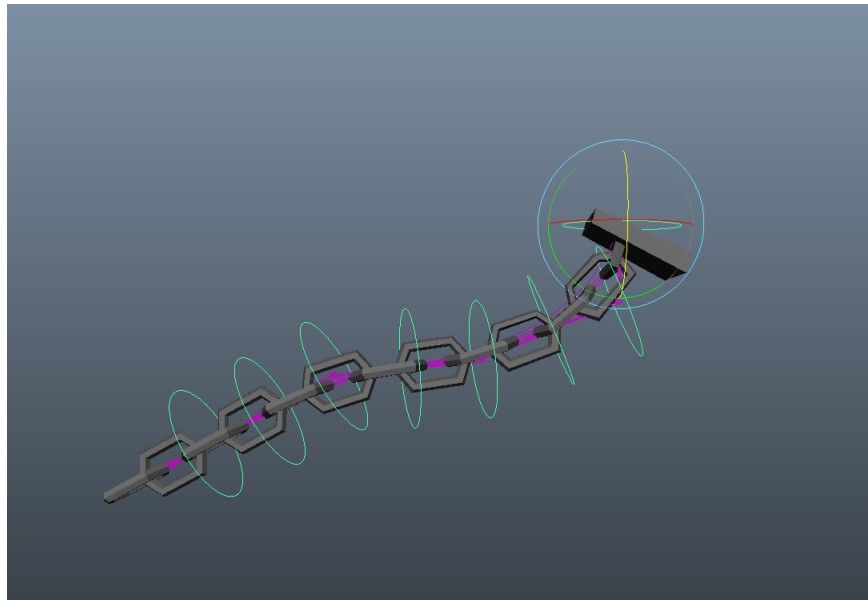


Figure 6.3: Using Hair Rig to Create Dynamic Chains

## Chapter 7

# Final Results

The final results of what the dynamic hair-rigging tool produces is displayed in figure 7.1. The figure shows the joint chain, the hair system it is suppose to follow, the FK joint chain with corresponding points to which positions are those of the FK controls specified by the user , and the control curves that maneuver the positioning of the joint chain. However the rig is posed, depending on the value of the curve attraction the hair system will always try to go back to that pose.

To test durability of this rigging tool the rig was applied to polygon hair strands on the rigged model from MeryProject.com shown in figures 7.2 and 7.3 [6]. For this test a very rough animation was made in order to show the dynamics of the rig. The rigging was produced on a select few of the hair curves. Though this tool can be very useful, there was still the long process of iterations. Depending on the use of the rigging system, the values of collision, curve attract, stiffness, positioning, drag, etc. will still need to be tweaked to the user's preference.

The dynamic hair-rigging system not only allows rigging artists to quickly and efficiently create rigs, it also provides the artistic intent of hair styles and animation. The rigging system allows an artist to adjust almost every aspect of hair such as shape, curl, bounce, stiffness, and collisions. Having the ability to add a rig to previously created hair systems, the rigging system provides an artist with more artistic freedom over the style and control over hair styles and other objects.



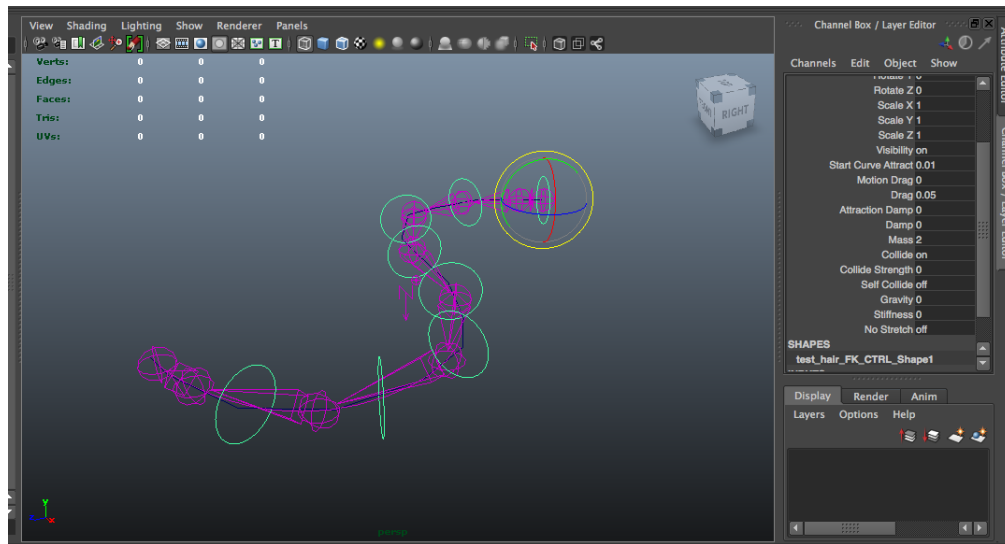


Figure 7.1: Hair Rig Bare Bones



Figure 7.2: Hair Rig on Mery Project Model

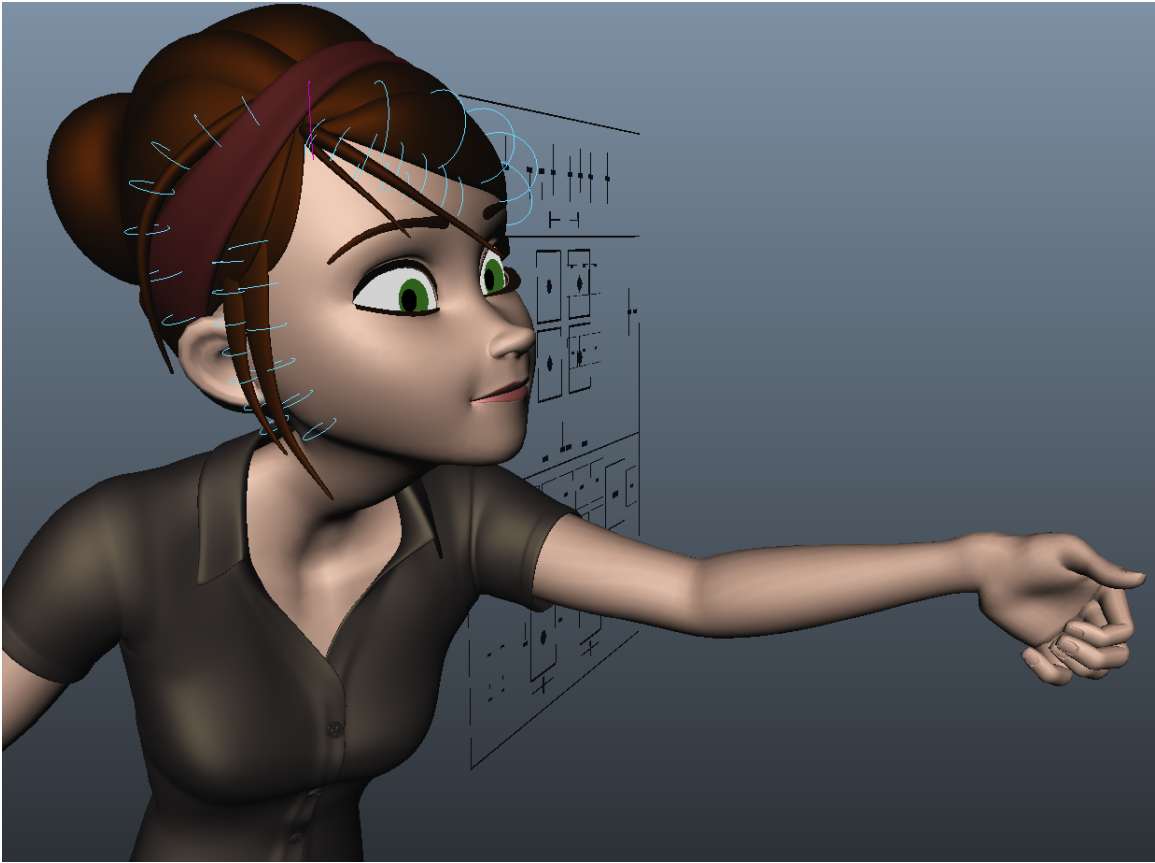


Figure 7.3: Second pose of Mery with hair rig

## Chapter 8

# Future Applications/Add Ons

Some additional functionality that could be added to this tool would be more groom control. When using this tool on a hair style, the user could choose one of the curves that are pre-added or create a new curve for hair to follow. If a new curve is created, using the tool and the current hair system, hairs would be added to the curve along with the controllers. The down side of this would be the lack of overall control and functionality. To further this tool, finding a way to have control over manipulating multiple curves at one time would be beneficial so clumps or sections of hair could be controlled or styled. Another feature that would be a beneficial addition is the use of multiple hair systems. The tool is able to create multiple hair systems, but the problem lies in the fact only one hair system and passive collider could interact with each other. If a user created a separate hair system for a character's bangs, or ponytail, the user could only chose one of those to interact with the character. Having to use only one hair system per passive collider causes a lack luster effect of the hair (or chain/rope) and seems too uniform and synchronized.

The image below, figure 8.1 illustrates how the tool could be used with a full hair system. From this scene, a curve was added to create a path for a bang that was added on the character's head. Once the bang was created, the tool was used to add the current hair system to the curve along with the controls.

Another feature that could be added is manipulation of the general shape of the hair. An example of how the control curves could look is shown in figure 8.2. After the user creates the rig, not only could the user change the position of the curve for the hair as well as the curvature, but to also be able to scale the hair as well. The control curves could manipulate multiple guide curves at

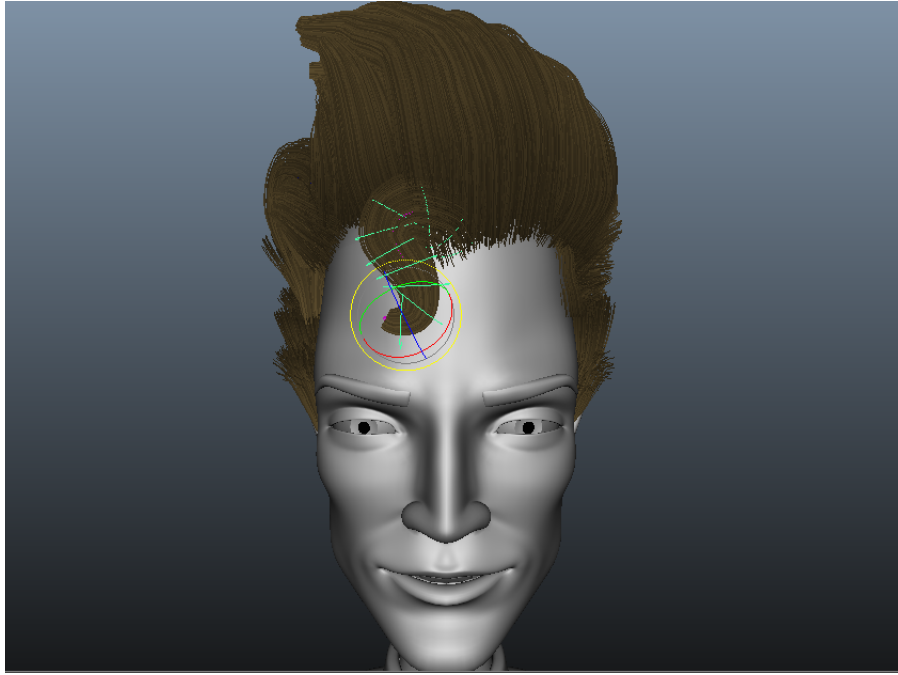


Figure 8.1: Using Tool to Create New Locks of Controllable Hair

one time, so the overall hair can be shaped or styled as well as acting like lattices to deform the hair or other object. Having this could help change the shape of the hair for particular scenes if need be.

A different kind of feature this tool could possibly be used for is appendages on creatures. This could range from antennae, tails, trunks, ears, etc. The usefulness of this would be the ability to use the dynamics and ability to key frame the controls to produce the best animation. Most of these appendages listed are controllable by the creature/animal but also have a constant motion to them due to momentum. These aspects could be achieved with using dynamic hair-rigging setup.

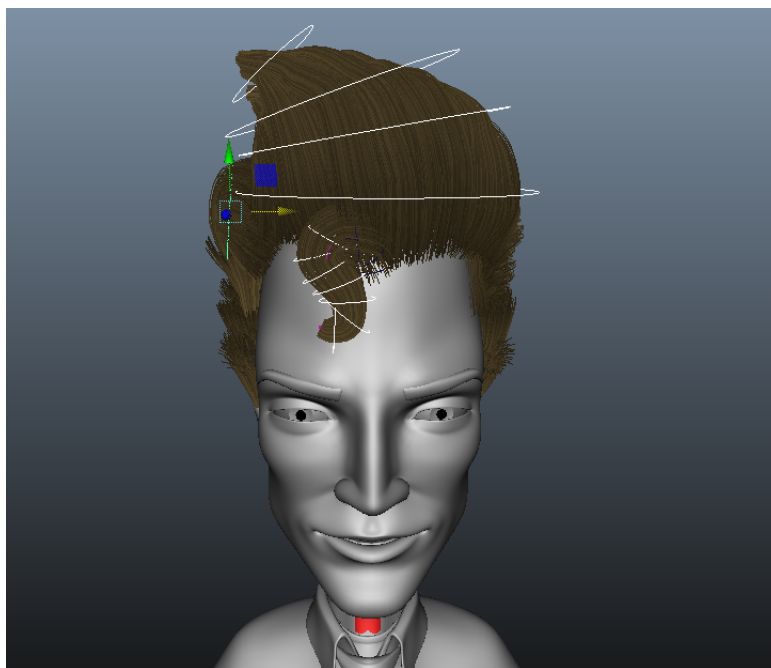


Figure 8.2: Possibility of having controls influence a wide range of hair by controlling more than one curve.

## Chapter 9

# Conclusion

The dynamic hair-rigging system includes an interface for the use of rigging, animating, and styling hair and other features. The interface allows users to easily create procedural rigs to aid in the posing and styling of hair. The rigging system also allows users to customize parameters quickly and efficiently by having one controller contain the properties of the entire rig. Styling certain characters's hair through rigging would not be as easy without the use of the rigging system. Placing joints and controls manually would not have been as accurate, as fast, or consistent as using the interface provided by the dynamic hair-rigging system. The rigging system also allows for the rig to be created using an existing hair system in Maya.

Functionality that the rigging system neglects includes skinning joints to mesh, creating different control shapes for rigging purposes, and the ability to replace the hair system. The ability to create different controllers would allow an animator to distinguish the purpose of each rig. The ability to replace the hair system would be a more elegant solution to starting from scratch if the first hair system needed to be replaced.

Procedural or automated rigging systems have become a staple for production-quality rigs in animation. The dynamic hair-rigging system provides an interface to aid rigging artists to quickly produce rigs to style and animate hair, chains, ropes, and more. The hair-rigging system is still a work in progress, as with most tools used in visual effects and animation. Further work on this project could expand use of the rigging system to Maya's other features such as fur, nCloth, multiple hair systems, and collisions between multiple hair systems.

# Appendices

# Appendix A

## Hair Rigging Code Snippets:

### .1 Creating Joints

```
def JointsOnCurve(self , event):  
  
    sel = mc.ls(sl=True)  
    curves = sel[0]  
  
    lcr = mc.spaceLocator()  
    loc = lcr[0]  
  
    path = mc.pathAnimation(loc , etu = 1000, f=True, c=curves)  
    numbers = mc.intSliderGrp("numberInt", q = True, v = True)  
  
    locPos = mc.xform(loc , q = True, ws = True, t = True)  
    divFactor = 1000/numbers  
  
    mc.select(cl=True)
```



```

i = mc.currentTime(q = True)

for i in range(1,1000,divFactor):

    mc.currentTime(i)
    lcPos = mc.xform("locator1", q=True, ws=True, t=True)
    joints = mc.joint(p=(lcPos[0], lcPos[1], lcPos[2]), a=True)

    selJoint = mc.ls(sl=True)
    jnt = selJoint[0]

    i = 1
    for i in range(numbers+1):
        mc.pickWalk(d='up')

    selectShape = mc.radioButtonGrp("radBtnGrpJoints", q = 1, sl = 1)
    #jointList = []
    if selectShape == 1:
        mc.joint(e = True, zso=True, oj = 'xyz', secondaryAxisOrient = 'zup', ch = 1)

    selectUniversal = mc.radioButtonGrp("radBtnGrpUni", q = 1, sl = 1)

    if selectUniversal == 1:
        mc.joint(e = True, zso=True, dof = 'xy', oj = 'xyz',
            secondaryAxisOrient = 'zup', ch = 1)

    elif selectUniversal == 2:

```

```

mc.joint(e = True, zso=True, dof = 'zy', oj = 'xyz',
secondaryAxisOrient = 'zup', ch = 1)

elif selectUniversal == 3:
    mc.joint(e = True, zso=True, dof = 'xz', oj = 'xyz',
secondaryAxisOrient = 'zup', ch = 1)

selectHinge = mc.radioButtonGrp("radBtnGrpHinge", q = 1, sl = 1)

if selectHinge == 1:
    mc.joint(e = True, zso=True, dof = 'x', oj = 'xyz',
secondaryAxisOrient = 'zup', ch = 1)

elif selectHinge == 2:
    mc.joint(e = True, zso=True, dof = 'y', oj = 'xyz',
secondaryAxisOrient = 'zup', ch = 1)

elif selectHinge == 3:
    mc.joint(e = True, zso=True, dof = 'z', oj = 'xyz',
secondaryAxisOrient = 'zup', ch = 1)

mc.delete(path)
mc.delete(loc)
mc.currentTime(1,e=True)

```

## .2 Adding Default Values to Hair System Parameters

```
def setDynamicParameters(self, dynItemsList, selectedHairSystem):  
  
    if selectedHairSystem == "":  
        hairSystRaw = dynItemsList[0]  
        hairSyst = mc.pickWalk(hairSystRaw, d="down")[0]  
        mc.setAttr(hairSyst + ".startCurveAttract", 0.35)  
        mc.setAttr(hairSyst + ".damp", 1)  
        mc.setAttr(hairSyst + ".drag", 0.05)  
        mc.setAttr(hairSyst + ".mass", 1)  
        mc.setAttr(hairSyst + ".attractionScale[0].attractionScale_FloatValue", 0.6)  
        mc.setAttr(hairSyst + ".attractionScale[1].attractionScale_FloatValue", 0.16)
```

### .3 Adding Hair System Attributes to the FK Control

```
def hairSystemAttributes(self, firstFkcontrol, dynamicItem, selectedHairSystem):

    if selectedHairSystem == "":
        hairSystRaw = dynamicItem[0]
        follicleRaw = dynamicItem[3]
        follicle = mc.pickWalk(follicleRaw, d="up")[0]
        hairSyst = mc.pickWalk(hairSystRaw, d="down")[0]
        startCurveAttraction = mc.checkBoxGrp("cbGrpDynAttrRow1", q=1,v1=1)
        motionDrag = mc.checkBoxGrp("cbGrpDynAttrRow1", q=1,v2=1)
        drag = mc.checkBoxGrp("cbGrpDynAttrRow1", q=1,v3=1)
        attractionDamp = mc.checkBoxGrp("cbGrpDynAttrRow1", q=1,v1=1)
        damp = mc.checkBoxGrp("cbGrpDynAttrRow2", q=1,v2=1)
        mass = mc.checkBoxGrp("cbGrpDynAttrRow2", q=1,v3=1)
        collide = mc.checkBoxGrp("cbGrpDynAttrRow3", q=1, v1=1)
        collideStrength = mc.checkBoxGrp("cbGrpDynAttrRow3", q=1, v2=1)
        selfCollide = mc.checkBoxGrp("cbGrpDynAttrRow3", q=1, v3=1)
        gravity = mc.checkBoxGrp("cbGrpDynAttrRow4", q=1, v1 = 1)
        stiffness = mc.checkBoxGrp("cbGrpDynAttrRow4", q=1, v2=1)
        noStretching = mc.checkBoxGrp("cbGrpDynAttrRow4", q=1,v3=1)
        PointLock = mc.checkBoxGrp("cbGrpDynAttrRow5", q=1, v1=1)

        default = startCurveAttraction + motionDrag + drag + attractionDamp + damp
        + mass + collide + collideStrength + selfCollide + gravity + stiffness
        + noStretching + PointLock

    if default ==0:
        mc.warning("all dynamic parameters selected, all are added by default")
```

```

if(startCurveAttraction ==1) or (default == 0):
    mc.select(firstFkcontrol , r = 1)
    mc.addAttr(ln = "StartCurveAttract", at = "float", min = 0, max = 1,
    dv = 0.35, keyable = 1, hnv = 1, hxv = 1)
    mc.connectAttr(firstFkcontrol + ".StartCurveAttract", hairSyst +
    ".startCurveAttract", f =1)

if(motionDrag == 1) or (default == 0):
    mc.select(firstFkcontrol , r = 1)
    mc.addAttr(ln = "MotionDrag", at ="float", min = 0, max = 1, dv = 0,
    keyable = 1, hnv = 1, hxv = 1)
    mc.connectAttr(firstFkcontrol + ".MotionDrag", hairSyst + ".motionDrag",
    f=1)

if(drag == 1) or (default == 0):
    mc.select(firstFkcontrol , r = 1)
    mc.addAttr(ln = "Drag", at="float", min = 0, max = 1, dv=0.05, keyable=1,
    hnv=1, hxv=1)
    mc.connectAttr(firstFkcontrol + ".Drag", hairSyst + ".drag", f=1)

if(attractionDamp ==1) or (default == 0):
    mc.select(firstFkcontrol , r = 1)
    mc.addAttr(ln = "AttractionDamp", at="float", min = 0, max = 1, dv=0,
    keyable=1, hnv=1, hxv=1)
    mc.connectAttr(firstFkcontrol + ".AttractionDamp", hairSyst +
    ".attractionDamp", f=1)

if(damp == 1) or (default == 0):
    mc.select(firstFkcontrol , r = 1)
    mc.addAttr(ln = "Damp", at="float", min = 0, max = 10, dv=2, keyable=1,

```

```

hmv=1, hmv=1)
mc.connectAttr(firstFkcontrol + ".Damp", hairSyst + ".damp", f=1)

if(mass == 1) or (default == 0):
    mc.select(firstFkcontrol, r = 1)
    mc.addAttr(ln = "Mass", at="float", min = 0, max = 10, dv=1, keyable=1,
    hmv=1, hmv=1)
    mc.connectAttr(firstFkcontrol + ".Mass", hairSyst + ".mass", f=1)

if(collide ==1) or (default ==0):
    mc.select(firstFkcontrol, r=1)
    mc.addAttr(ln = "Collide", at="bool", keyable=1)
    mc.connectAttr(firstFkcontrol + ".Collide", hairSyst + ".collide", f=1)

if(collideStrength == 1) or (default == 0):
    mc.select(firstFkcontrol, r=1)
    mc.addAttr(ln="CollideStrength", at = "float", min = 0, max =10, dv=0,
    keyable=1, hmv=1, hmv=1)
    mc.connectAttr(firstFkcontrol + ".CollideStrength", hairSyst +
    ".collideStrength", f=1)

if(selfCollide ==1) or (default == 0):
    mc.select(firstFkcontrol, r=1)
    mc.addAttr(ln="SelfCollide", at = "bool", keyable=1)
    mc.connectAttr(firstFkcontrol + ".SelfCollide", hairSyst + ".selfCollide",
    f=1)

if(gravity ==1) or (default ==0):
    mc.select(firstFkcontrol, r=1)

```

```

mc.addAttr(ln="Gravity", at = "float", keyable=1, dv = 0.98, hnv=1, hxv=1)
mc.connectAttr(firstFkcontrol + ".Gravity", hairSyst + ".gravity",f=1)

if(stiffness ==1) or (default == 0):
    mc.select(firstFkcontrol , r=1)
    mc.addAttr(ln="Stiffness", at = "float", keyable=1, dv = 0, min = 0,
    max = 10, hnv=1, hxv=1)
    mc.connectAttr(firstFkcontrol + ".Stiffness", hairSyst + ".stiffness",f=1)

if(noStretching == 1) or (default == 0):
    mc.select(firstFkcontrol , r = 1)
    mc.addAttr(ln = "NoStretch", at="bool", keyable=1)
    mc.connectAttr(firstFkcontrol + ".NoStretch", hairSyst + ".noStretch",
    f=1)

if(PointLock == 1) or (default == 0):
    mc.select(firstFkcontrol , r = 1)
    mc.addAttr(ln = "PointLock",enumName="NoAttach=0:Base=1:Tip=2:BothEnds=3"
    , keyable=1)
    mc.connectAttr(firstFkcontrol + ".PointLock", dynamicItem[3]
    + "Shape.pointLock",
    f=1)

```

# Bibliography

- [1] Autodesk. hairsystemshape (nhair) node. <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/Maya/files/Hair-Nodes-hairSystemShape-htm.html>, 2014.
- [2] Autodesk. Joints and bones. <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-1B59334F-2605-44C3-B584-A55B239A2CBE-htm.html>, 2016.
- [3] Autodesk. nhair overview and concepts. <https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/Maya/files/GUID-70A85F9C-9FC0-49FD-A535-76D379D2C9BC-htm.html>, 2016.
- [4] Family Choice Awards. Mr. peabody and sherman. <http://www.familychoiceawards.com/entertainment/mr-peabody-sherman-blu-raydvd-digital-hd-combo-pack/>, 2015.
- [5] Oliver Franklin-Wallis. How disney created the ludicrously realistic fur of zootropolis. <https://www.wired.co.uk/news/archive/2016-03/22/disney-zootropolis-fur-animation>, 2016.
- [6] Antonio Mendez Lora Jose Manuel Garcia Alvarez. Merry project: free character for animators. [www.merryproject.com](http://www.merryproject.com), 2014.
- [7] Mona Lalwani. Fur technology makes zootopia’s bunnies believable. <http://www.engadget.com/2016/03/04/fur-technology-makes-zootopias-bunnies-believable/>, 2016.
- [8] Mark Masters. Key 3d rigging terms you need to know. <http://blog.digitaltutors.com/key-rigging-terms-get-moving/>, 2014.
- [9] Pixar. Brave new hair. <https://www.fxguide.com/featured/brave-new-hair>, 2015.
- [10] Ben Sledge. An automated build system for articulated characters. Master’s thesis, Clemson University, May 2014.
- [11] Chris Zurbrigg. Display color override. <http://zurbrigg.com/maya-scripts/display-color-override>, 2012.